FBSNG – Batch System for Farm Architecture

J.Fromm, K. Genser, T. Levshina, I. Mandrichenko
Farms and Clustered Systems Group,
Computing Division,
Fermilab

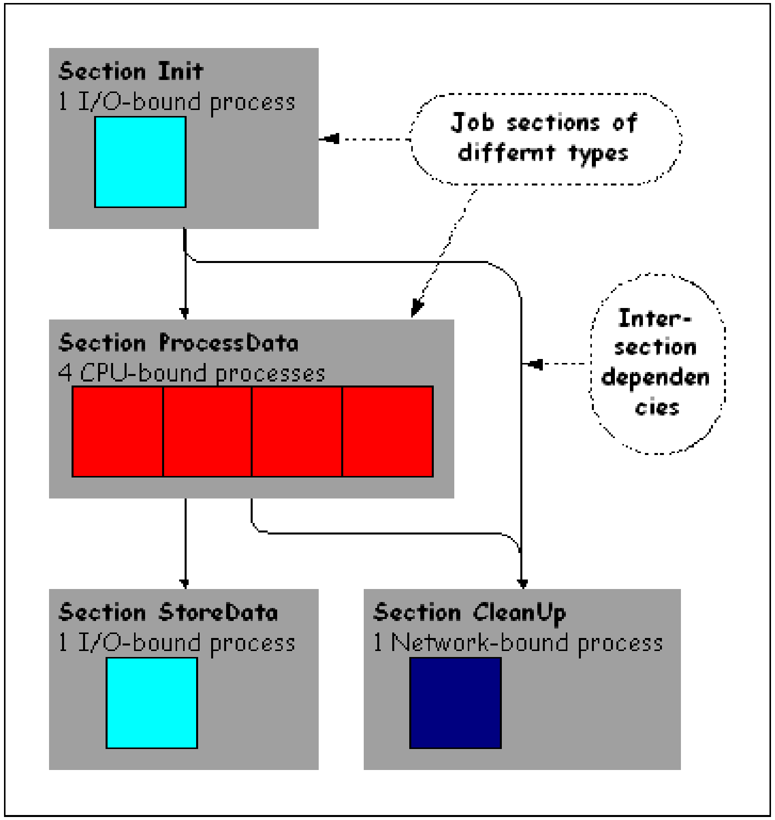
Introduction and Project History

- FBSNG (and its predecessor FBS) is a batch system designed as a resource management tool for computing farms used for RunII off-line data processing at Fermilab.
- Typical Run II farm is expected to consist of ~150-300 800MHz-1GHz dual-Pentium CPU computers.
- FBSNG is designed to manage computing farms of up to 1000 nodes
- Project history:
  - Spring 1998 – initial FBS design, first working prototypes
  - Fall 1998 – first production users (E871)
  - Fall 1999 – FBS v2.2 (last FBS version) released
  - Fall 1999 – beginning of FBS redesign project (FBSNG)
  - July 2000 – FBSNG v1.0 released into production
  - Summer 2000 – FBS is replaced by FBSNG
  - June 2001 – FBSNG v1.3 released

FBSNG Concepts: Specifics of Farm Architecture

- Typical computing farm consists of large (~10-~1000) number of “small” computers
- Typical farm computer has limited resources
  - 1-2 (-4?) CPUs
  - Limited disk capacity 10-50 GB
  - 1 NIC 100 Mbit/sec
- Therefore, each computer can run 1-2 (-4?) CPU-bound processes
- Resource Counting concept:
  - Know resource capacity of each farm node
  - Know process resources requirements
  - Know which process runs on which node
  - Start new process when and where resources are available
- *Resource counting is much simpler than load measuring, yet it is sufficient for load averaging on farms*

FBSNG Concepts: FBSNG Job Structure



- FBSNG *Job* consists of one or more *Sections*
- Each Section is an array of “identical” batch processes
  - Simplest job: 1 section of 1 process
- All section processes start at the same time
- They may or may not cooperate through IPC (MPI, PVM, FIPC, etc.)
- Inter-section dependencies, e.g.:
  - **Process data** only after **Init** succeeds
  - Perform **CleanUp** if either **Init** or **ProcessData** fails

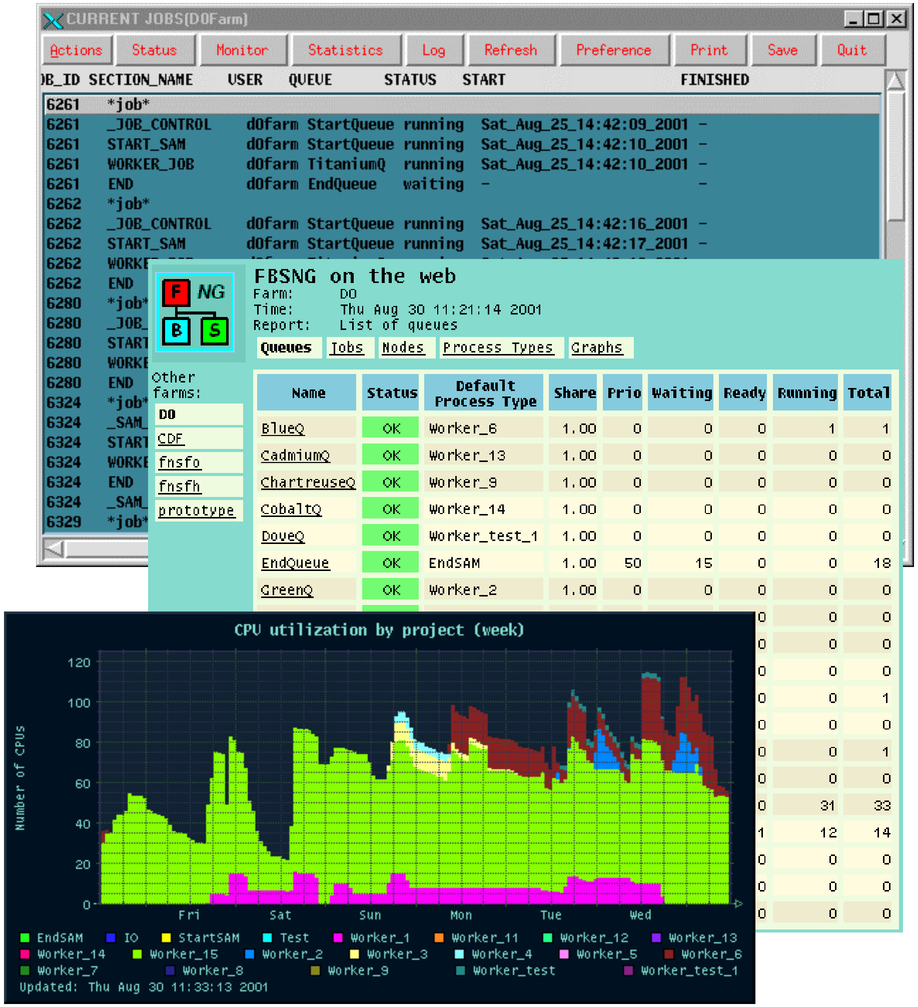
FBSNG Concepts: Abstract Resources

- FBSNG resources are just counters. There is no presumption that they represent any real computational resources.

  Hence Abstract Resources
- Global resources - visible to the entire farm
  - Examples: Disk space on NFS server, network bandwidth
- Local resources - Can be used only by processes running on the node
  - Examples: CPU, local scratch disk
- Node Attributes
  - Can be viewed as local resources with unlimited capacity.
  - Examples:
    - Special software installed (e.g. scientific library, version of OS)
    - … or even computer case color
  - Can be used to *logically* partition the farm into smaller “subfarms”
- Resources can be created and removed *dynamically* at any time
- Interchangeable resources can be combined into *resource pools:*
  - I need 6GB of space on whichever disk is available

User Interface: Overview

- Command line interface
  - Job submission, control, monitoring
  - Resources, Scheduler monitoring
  - Administrator’s tools
- GUI
  - Job control, monitoring
  - Resources statistics
  - Scheduler monitoring
  - Administrator’s tools
- Web interface (FBSWWW)
  - Job monitoring
  - Resources statistics
  - Scheduler monitoring
- Python API
  - All of the above plus asynchronous job status change notifications
  - UI, GUI, FBSWWW use API



```
B_ID SECTION_NAME     USER   QUEUE      STATUS   START                     FINISHED
6261   *job*
6261   _JOB_CONTROL     d0farm StartQueue running  Sat_Aug_25_14:42:09_2001 -
6261   START_SAM        d0farm StartQueue running  Sat_Aug_25_14:42:10_2001 -
6261   WORKER_JOB       d0farm TitaniumQ  running  Sat_Aug_25_14:42:10_2001 -
6261   END              d0farm EndQueue   waiting  -                        -
6262   *job*
6262   _JOB_CONTROL     d0farm StartQueue running  Sat_Aug_25_14:42:16_2001 -
6262   START_SAM        d0farm StartQueue running  Sat_Aug_25_14:42:17_2001 -
6262   WORKE
6262   END
6280   *job*
6280   _JOB_
6280   START
6280   WORKE
6280   END
6324   *job*
6324   _SAM_
6324   START
6324   WORKE
6324   END
6324   _SAM_
6329   *job*
```

FBSNG on the web

Farm: D0
Time: Thu Aug 30 11:21:14 2001
Report: List of queues

Other farms:

D0
CDF
fnsfo
fnsfh
prototype

Name	Status	Default Process Type	Share	Prio	Waiting	Ready	Running	Total
BlueQ	OK	Worker_6	1.00	0	0	0	1	1
CadmiumQ	OK	Worker_13	1.00	0	0	0	0	0
ChartreuseQ	OK	Worker_9	1.00	0	0	0	0	0
CobaltQ	OK	Worker_14	1.00	0	0	0	0	0
DoveQ	OK	Worker_test_1	1.00	0	0	0	0	0
EndQueue	OK	EndSAM	1.00	50	15	0	0	18
GreenQ	OK	Worker_2	1.00	0	0	0	0	0
						0	0	0
						0	0	0
						0	0	0
						0	0	1
						0	0	0
						0	0	1
						0	0	0
						0	31	33
						1	12	14
						0	0	0
						0	0	0
						0	0	0

CPU utilization by project (week)

Updated: Thu Aug 30 11:33:13 2001

User Interface: Job Submission

- Using Job Description File (JDF)
  - Describe job structure in JDF
  - Submit the job with

    ```
    fbs submit myjob.jdf
    ```
  - Full job description functionality
- Using one-line command
  - Simple 10 process job:

    ```
    fbs exec -q MyQueue -n 10 /home/user/runjob.sh
    ```
  - Allows to run batch job in interactive mode:

    ```
    fbs exec -q MyQueue -n 5 -I /bin/tcsh
    ```
  - Limited job description capabilities
- Using API – full access to job description and control functionality

- As a *configuration option*, FBSNG can use **Kerberos** v5 for client authentication
- If necessary, FBSNG creates Kerberos credentials for batch processes

FBSNG Scheduler

- Farm-aware: unit of scheduling is an array of batch processes (job section)
- Administrators controls:
  - Most flexible: Resource utilization shares can be assigned to projects
  - Projects can be limited by (abstract) resource utilization quotas
  - Least flexible: Farm can be partitioned into smaller farms, and projects can be confined to their sub-farms
- Guaranteed scheduling:
  - Regardless of competition, bulk job is guaranteed to start in finite amount of time according to defined project share.
  - Small jobs will be held if necessary.
- Scheduling parameters can be changed dynamically at any time.

Summary

- FBS and later FBSNG have been in production at Fermilab since 1998
- FBSNG has proven to be portable, simple, robust, flexible and powerful resource management tool for farms or clusters architecture.

- FBSNG has been successfully used to manage wide variety of computational projects such as
  - off-line data processing
  - Monte-Carlo generation

  on *dedicated* farms owned by single group of users such as CDF and D0

  as well as on farms *shared* by multiple groups.

- Currently, FBSNG is used on:
  - 2 common use (a.k.a. fixed target) farms at FNAL (~50 nodes each)
  - CDF, D0 farms (100+ nodes)
  - NIKHEF (D0 collaborators)
  - US CMS tier 1 site farm at FNAL
  - Other farms at HEP as well as non-HEP organizations